# Pre-Meeting Support Views based on SISCO Model

*Cavalcanti, M. C.*[1] *- Espinosa, J.*[2] *- Maroja, B.*[3] *- Endo, M. Y*[4]

yoko@nce.ufrj.br - jespinos@dcc.uchile.cl – brma@di.ufpe.br - yuly@nce.ufrj.br

## *Abstract*

*This report presents and compares three different approaches on the implementation of a pre-meeting support system. The prototypes described here were developed by three independent teams, two from Brazil and one from Chile. SISCO-Recife and SISCO-Rio are the Brazilian prototypes and USISCO is Chilean. The development of the three prototypes was based on the same SISCO discussion model. SISCO has as its main goal the enhancement of meetings productivity through previous discussion. The developed prototypes support asynchronous and distributed collaboration, as a preparation for a face-to-face meeting.*

**Keywords:** CSCW, meeting support system, client-server.

# 1. Introduction

The use of computer systems to intermediate and facilitate human interaction has been growing fast. CSCW - ***Computer Supported Cooperative Work*** research area examines how computers may support people working together. CSCW applications are called groupware which are designed to support a specific group task. In the literature there are many different groupware classifications. The space/time classification is considered the most traditional one. It categorizes groupware applications according to the group localization (same space/different space), and the synchronicity of the task (same time/different time). The space/time classification originated two other variations, considering previsibility and group size [Borges 95].

Another classification for groupware considers the application functionality [Ellis et al. 91] and provides a general idea of the breadth of the groupware domain, by creating categories based on the main software functionality, for example, message systems, co-authoring systems, discussion systems, electronic meeting systems, and so on. A message system is basically an electronic mailing manager. Co-authoring systems support the group creation of any object, like texts, drawings, schemes, etc. Conference systems are usually sophisticated electronic mailing systems. They provide a specific message structure, organizing them in groups (also called conferences), each one with different members and messages. Finally, there are the electronic meeting systems which can be classified in two other subcategories: Electronic Rooms, which provide the infrastructure for face-to-face group meetings, and Decision Support Systems (DSS), which are specifically designed to support the group decision on important issues.

According to these classifications, SISCO prototypes may be classified as asynchronous distributed conference systems. However we also consider them as electronic meeting systems because they indirectly support meetings. SISCO project research group, after analyzing some meeting problems, like uncertainty, equivocality [Daft & Lengel 86, Watson 93] and time consuming, proposed a pre-meeting system [Bellassai et al. 95]. The main objective of such system is the meeting participants preparation through previous asynchronous discussion, resulting in a much more productive and qualitative face-to-face meeting. It is important to notice that the proposed system is not meant to support

[1] Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil - NCE/UFRJ

[2] Departamento de Ciencias de la Computación, Universidad de Chile, Santiago, Chile - DCC/UChile

[3] Departamento de Informática, Universidade Federal de Pernambuco, Recife, Brazil - DI/UFPE

[4] Departamento de Ciência da Computação, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil - DCC/IM/UFRJ

the meeting itself and therefore, it is not aimed to support the decision process. The prototypes described here are the first implementation of the SISCO system proposal.

Partially funded by an Ibero-American program (CYTED)[5], SISCO project research group is formed by members of different countries. SISCO prototypes were developed by three teams: two of them are from Brazil and the third one is from Chile. The Brazilian prototypes were baptized with the name of the cities they were developed at: SISCO-Recife and SISCO-Rio de Janeiro. Both of them were developed at federal universities: Universidade Federal de Pernambuco (UFPE) and Universidade Federal do Rio de Janeiro (UFRJ). The Chilean prototype, USISCO was developed at Universidad de Chile in Santiago, Chile.

This report is organized in five sections. The next one presents the SISCO data model. Section 3 presents the description of each prototype: USISCO, SISCO-Recife and SISCO-Rio. Section 4 discusses the prototypes implementation and functionality. Finally, section 5 concludes the report.

## 2. SISCO model

A constructive discussion consists of presenting ideas in an organized way, and of helding the meeting in an environment that provides the necessary tools for the participants to check the progress of the meeting. Based on this idea, some authors had proposed discussion and argumentation models. Most of them suggest ways for classifying the discussion contributions into categories, and implemented it using natural language editors.

In 1970, Kunz and Rittel proposed a model called IBIS (Issue Based Information System) [Kunz et al. 70] composed of 3 elements and 9 relationships between the elements. The IBIS model represents the main elements of a discussion, allowing its understanding and easy contribution. Because of its simplicity and its intuitive use, the model has been successfully used [Conklin et al. 88] (see figure 2.1 below).

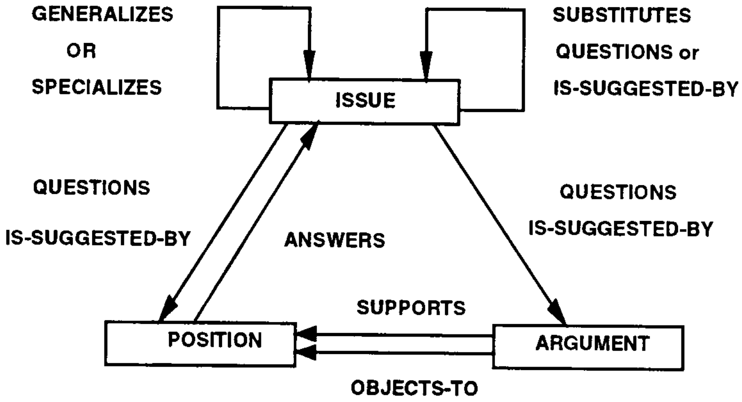


*Figure 2.1: IBIS Model*

It is important to notice that in IBIS model the participants are restricted to three elements and nine kinds of relationships. In order to attend the SISCO project specification [Bellassai et al. 95], it was necessary to extend the IBIS model, adding new elements and relationships to it. The new elements are described below, followed by a figure that illustrates the new model architecture (see figure 2.2).

**Participant**: This element was created due to the need of identifying the origin of the statements. However, the anonymous contributions should also be supported. It was defined a specialization of a participant related to the role of meeting coordinator.

[5] SISCO project – CYTED program VII.6

**Pre-Decision**: The pre-decision element was created because of the importance of representing to previous decisions or assertions, or even some meeting guidelines. Pre-decisions can in some cases take the format of constraints.

**Proposal**: This is a specialization of the Issue element. It suggests an action, such as a task or an Issue redefinition.

**Task**: As a result of a discussion, some extra documents or additional information might be identified as important to endorse or to help an argument. The task elements represent the needed actions for supplying the identified items.

**Remark**: Within a discussion there might be elements which do not fit in the IBIS elements. In order to avoid the distortion of the basic elements, it was created the remark element.

For a more detailed description of SISCO model and specification refer to Bellassai et al. 95.

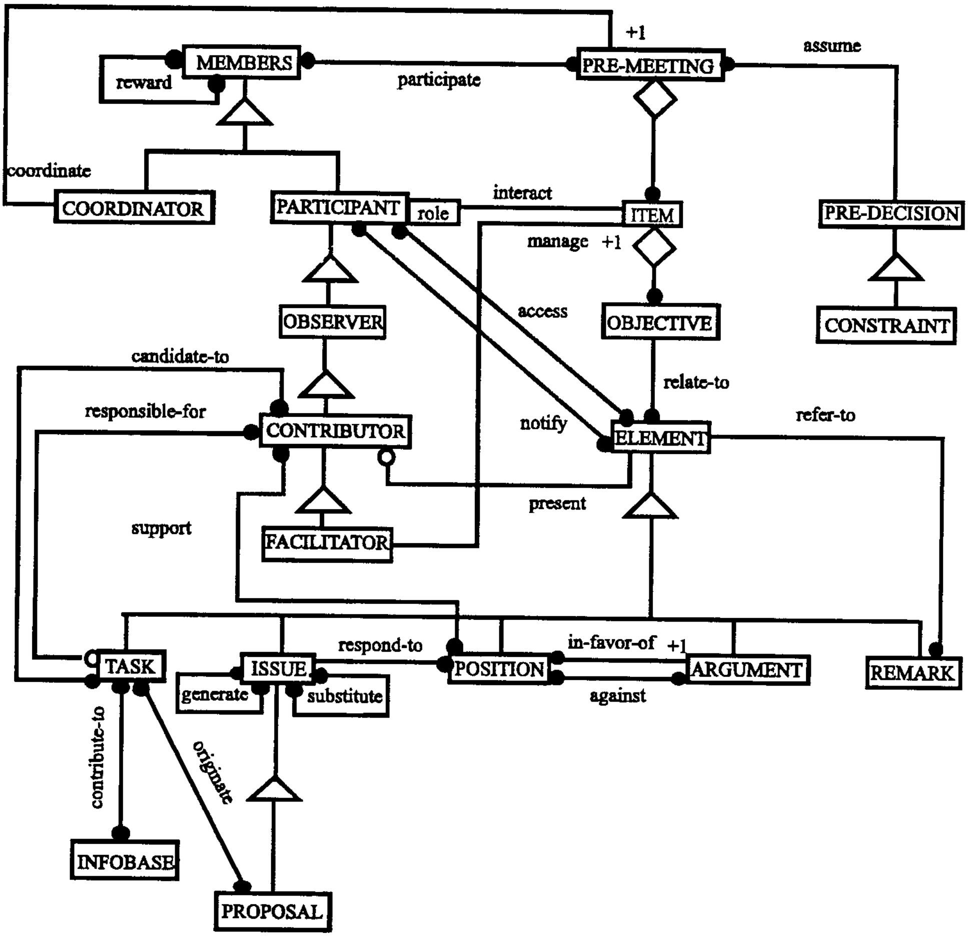


*Figure 2.2: SISCO Data Model*

## 3. Prototype Description

SISCO prototypes were developed independently. However, the three of them were based on two project premises. The first premise was that the prototype implementation should be based on the SISCO model. Second, in order to provide a high level of accessibility, the prototypes should be implemented over the Internet.

## *3.1. USISCO*

The USISCO prototype was developed at Universidad de Chile by Jaime Espinosa under Prof. Jose A. Pino supervision. The USISCO had as its main motivation the development of a product that could be portable to distinct platforms and that would use the TCP/IP as the communication protocol for connecting with the database server, which would not necessarily be in the same machine of the client application.

USISCO should be an user-friendly application for a regular user. Also, considering the interface aspect as a fundamental goal for USISCO, it should be implemented in a programming language or tool that could allow a powerful and standard graphic interface.

It was considered to implement USISCO over the Web system, in such way that the user would not be obligated to use a special architecture for executing the prototype. However, this option was discarded as for installing a USISCO server it would be necessary a Web master help. For this reason, Java programming language was the chosen option, allowing an easy and powerful programming of graphic interfaces and TCP/IP network connections.

By the time of the beginning of USISCO construction there was no possibility of connecting Java with robust Data Base Management Systems. At that time, **MiniSQL** was one of the few DBMS's which offered Java connections, through the use of a class packet called **Msql**. MiniSQL is a small database engine which was designed to provide a fast access to low quantities of stored data. MiniSQL offers a subset of the defined SQL potentialities, except for views, nested queries and transactions. Finally, MiniSQL is ANSI compliant and it was designed for TCP/IP connected client/server application development.

### 3.1.1. USISCO Model

In order to improve the interface and add some new functionality to the prototype, the SISCO original data model was altered. Figure 3.1 shows an Entity-Relationship representation of USISCO model. Some of the simplifications and extensions assumed in USISCO model are pointed out in the list below:

- For a more important and clear role in the discussion, the pre-decisions were associated to the meeting items.

- The **marked** concept was introduced, which means that the user may **mark** any discussion element, and return to them in later sessions by directly using a search facility.

- The original model did not provide the superuser concept, which would be a group of people with the special permission of creating new meetings. USISCO implements this concept by providing a database called SISCO which contains the superusers, for each MiniSQL server.

- The "new" and "read" concepts were also added, which enables each user to easily identify the new and unread discussion items in his/her session.

- The concept of "substituted" was extended to every discussion item, not only to issues.

- The concept of "meeting" was not implemented as part of a database. Considering MiniSQL easy database creation in different servers, each database was created to support a specific meeting.

- It was not implemented the issuc generation concept, nor the position supporting concept.

Even though USISCO had modified the original model, it did follow an important SISCO concept, which says that the discussion memory should not be altered. The other prototypes did not implement this concept, allowing the users to delete any element of a discussion.

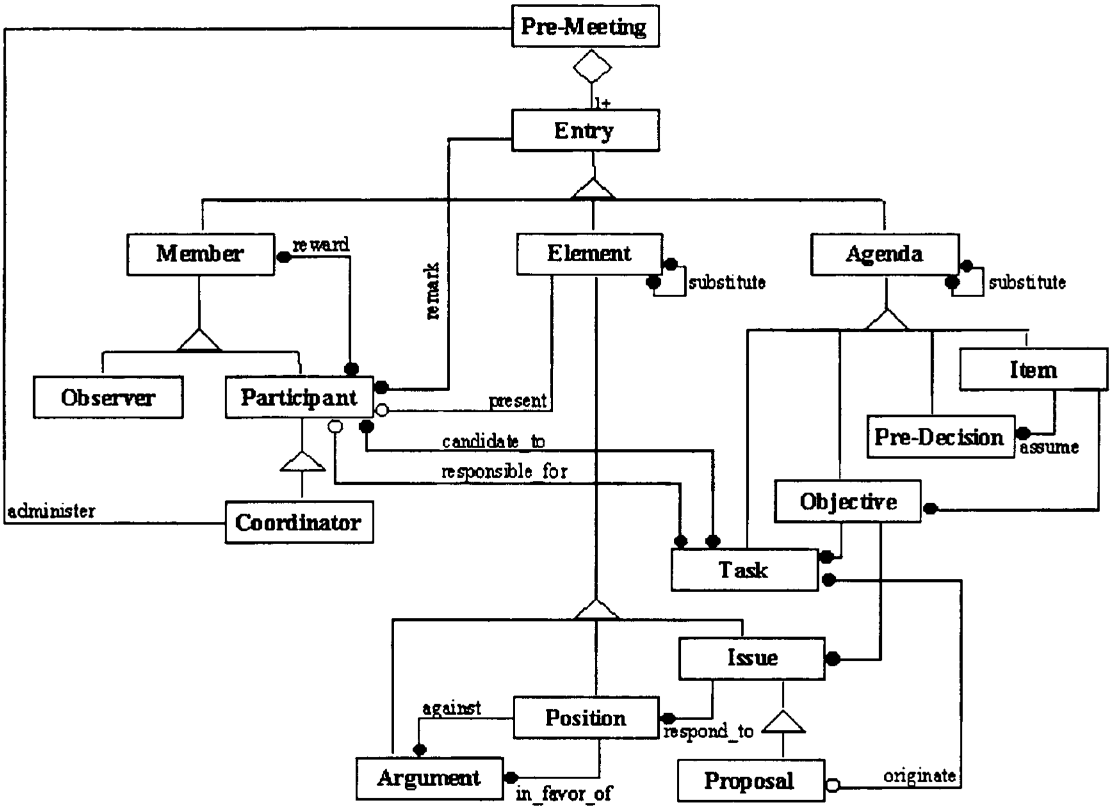


*Figure 3.1: USISCO Data Model*

### 3.1.2. USISCO Architecture

In order to run USISCO, the client side needs the Java Development Kit 1.0.2 or superior in a graphic environment. The server side needs a MiniSQL server version 1.0.16, which could be installed in SunOS 4.1.1 or superior, Solaris 2.3 or superior, Ultrix 4.3, Linux, OSF/1, HP-UX, NeXT, SCO, Cray, Tandem and others. The figure below illustrates the USISCO architecture. Even though USISCO was designed to be used as a client/server application, it is possible to use it in a single machine.

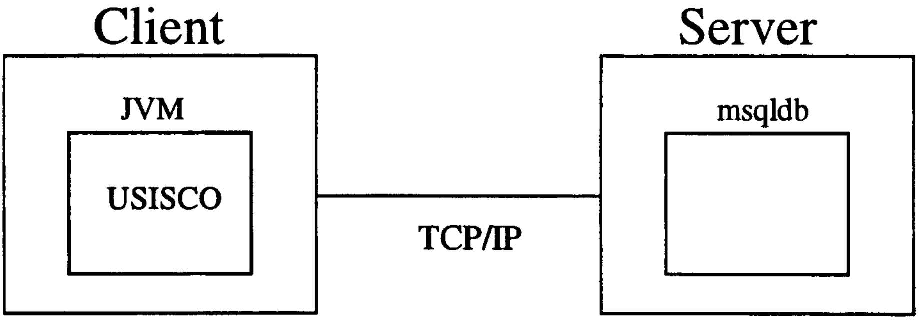


*Figure 3.2: USISCO architecture*

USISCO client side was developed in JAVA 1.0.2 under Linux 1.2.13 and under Solaris 2.3, locally accessing the MiniSQL server (in the same machine). Tests proved the client/server remotely connection to be working: the client side runs on a Solaris 2.3 and the server side (MiniSQL) runs on a SunOS 5.4.

### 3.1.3. USISCO Prototype

USISCO first window introduces itself to the user through a presentation window (see figure 3.3), from which he/she can choose between getting the product description (***About***) and administrating pre-meetings (***Continue***).

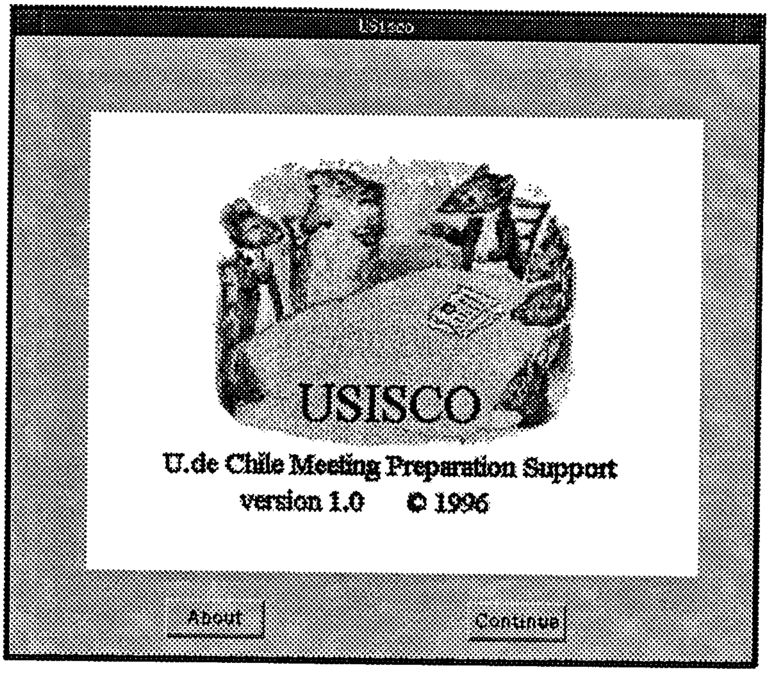
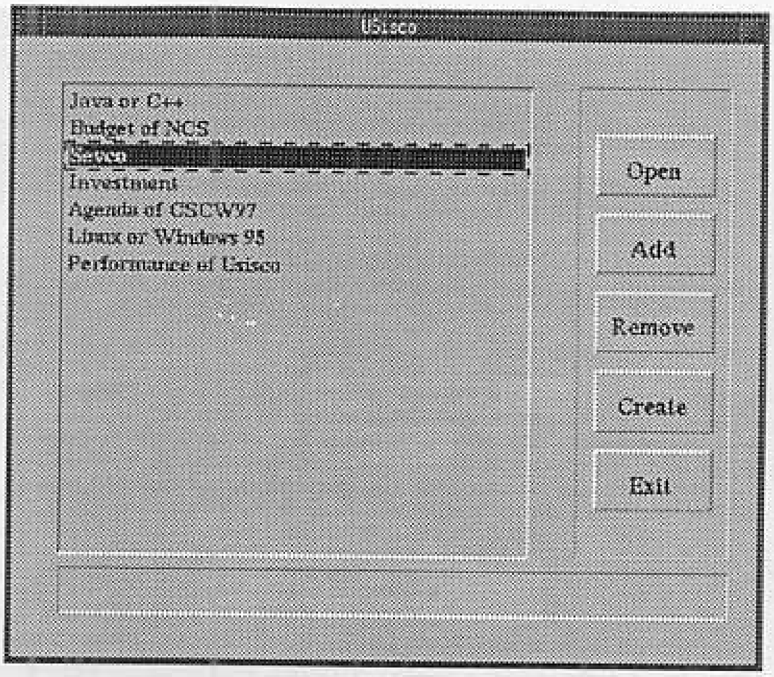


***Figure 3.3: USISCO Presentation Window***

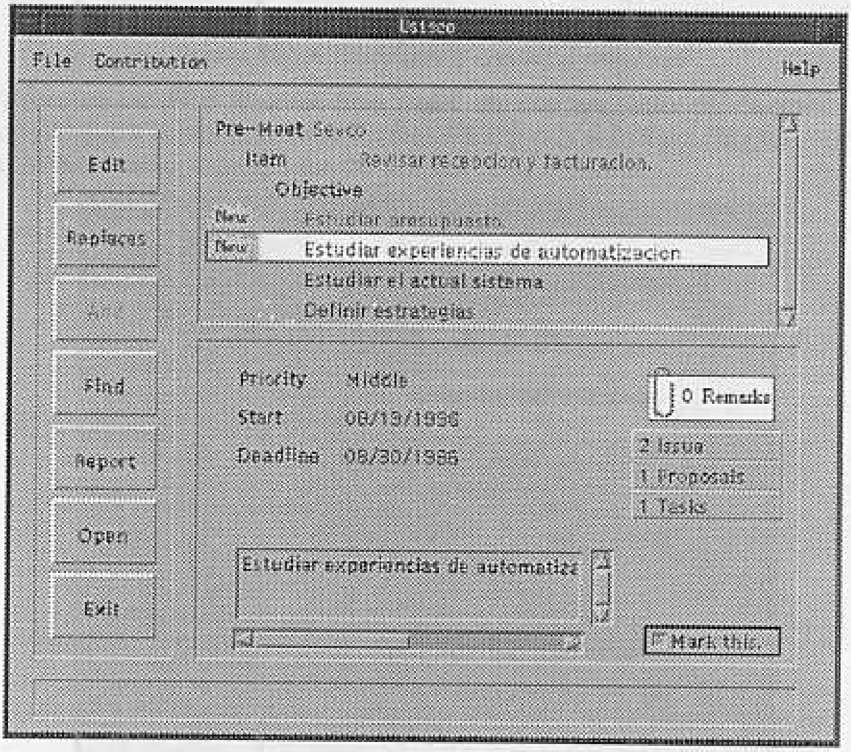
The pre-meeting administration window (see figure 3.4) provides a list of the pre-meetings in which the user has a particular interest. The user is able to configure this list by **adding** and **deleting** pre-meetings to it. The **create** button allows the creation of a new pre-meeting, which will be automatically **added** to the list. For each new pre-meeting, it is necessary to configure the proper permission in the server in which it will be stored. Finally, the user may choose to participate on a discussion by **opening** the correspondent pre-meeting.

The opening of a particular pre-meeting, causes the user's login/password authentication. Once authorized, the user gets a ***Discussion hierarchy Window*** (see figure 3.5), which provides all the actions allowed for that discussion, considering the user status: coordinator, participant or observer. Basically, it is possible ***editing***, ***replacing*** or ***adding*** elements to the discussion (see Figure 3.6). It is also provided a report functionality, which generates a list of selected elements and their correspondent hiearchy. Another interesting functionality is the ***find*** option, which allows a direct search for a particular element, avoiding the hierachical traversing. Finally, it is possible to open another pre-meeting, without returning to the previous window.



Java or C++
Budget of NCS
Sevco
Investment
Agenda of CSCW97
Linux or Windows 95
Performance of Usisco

*Figure 3.4: USISCO Administration of Pre-Meetings*



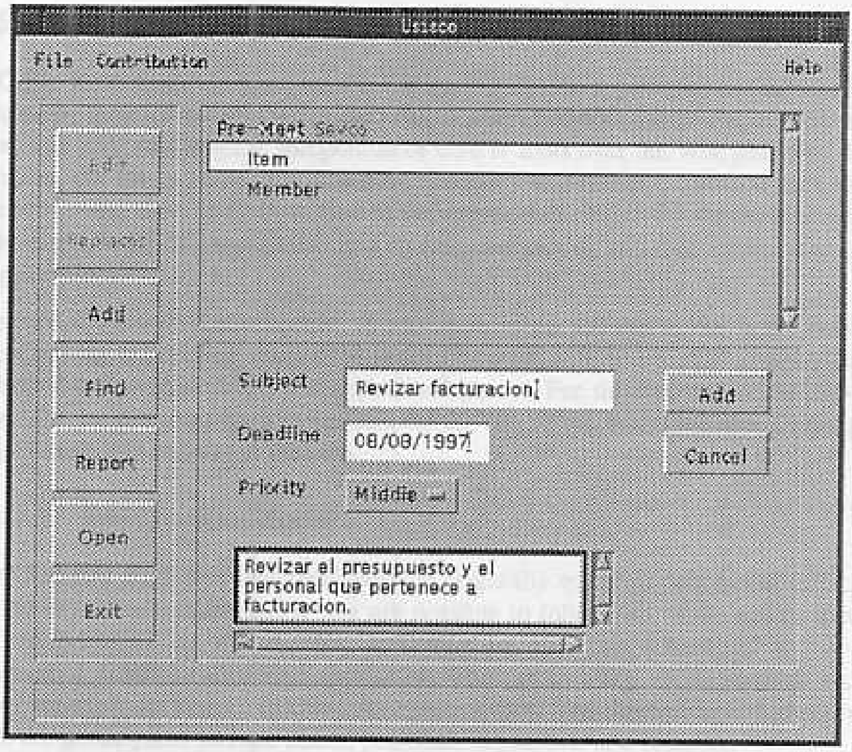
Pre-Meet Sevco
Item Revisar recepcion y facturacion.
Objective
New Estudiar presupuesto.
New Estudiar experiencias de automatizacion
Estudiar el actual sistema
Definir estrategias

Priority Middle
Start 08/13/1996
Deadline 08/30/1996

2 Issue
1 Proposals
1 Tasks

Estudiar experiencias de automatiza

*Figure 3.5: Discussion hierarchy Window*



Subject: Revizar facturacion

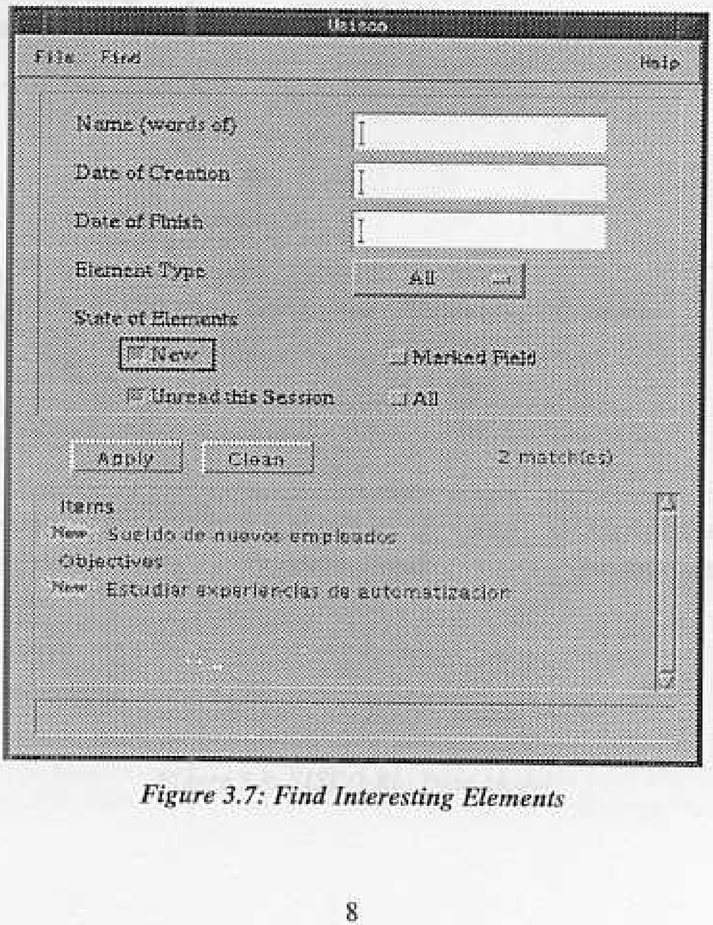
Deadline: 08/08/1997

Priority: Middle

Revizar el presupuesto y el personal que pertenece a facturacion.

*Figure 3.6: A Coordinator creates a new Item*

Figure 3.7 presents the ***Find Window***, from which the user may filter the discussion elements, filling the forms accordingly. It is important to notice that the user may search not only the ***New*** elements, but also the ***Unread in this Session*** elements and the ***Marked Field*** elements.



Name (words of):

Date of Creation:

Date of Finish:

Element Type: All

State of Elements

☒ New

☐ Marked Field

☒ Unread this Session

☐ All

2 match(es)

Items

New Sueldo de nuevos empleados

Objectives

New Estudiar experiencias de automatizacion

*Figure 3.7: Find Interesting Elements*

## 3.2. SISCO-Rio

All SISCO-Rio team members have been investigating CSCW applications even before the team establishment. Originally they joined another project called InterConnect [Campos et al. 95], in which a group of researchers has been studying the integration of CSCW tools over the Web platform. In order to provide this integration, the group realized it would be necessary to develop a common data base, where the various applications could share their data. One of the approaches chosen for the development of such data base, was to develop one CSCW application and its data base model, and then extend it to be the data resource to other applications.

When the SISCO project was proposed, it was seen as an answer to the InterConnect needs. Therefore, the InterConnect researchers decided to share resources with the SISCO project, providing a team specially dedicated to the SISCO-Rio prototype development. For this reason, the SISCO-Rio team focused on the development of a data base model and on the use of the Web platform, investigating the integration of this platform with relational DBMSs.

### 3.2.1. SISCO-Rio Model

The SISCO-Rio prototype was developed over the originally specified data model (see chapter 2). Even though we tried to follow it faithfully, it was not possible to follow the whole model specification. We established some priorities on the entities, according to what we considered to be the basic functionality: pre-meeting administration and discussion. The pre-meeting administration includes the creation of a new pre-meeting, its items, objectives and participants. The discussion includes pre-meeting participation, where the participants include issues, positions, arguments and remarks.

Task, Infobase, Pre-Decision and Constraint are the left out entities of the original model. The figure below shows the simplified model.

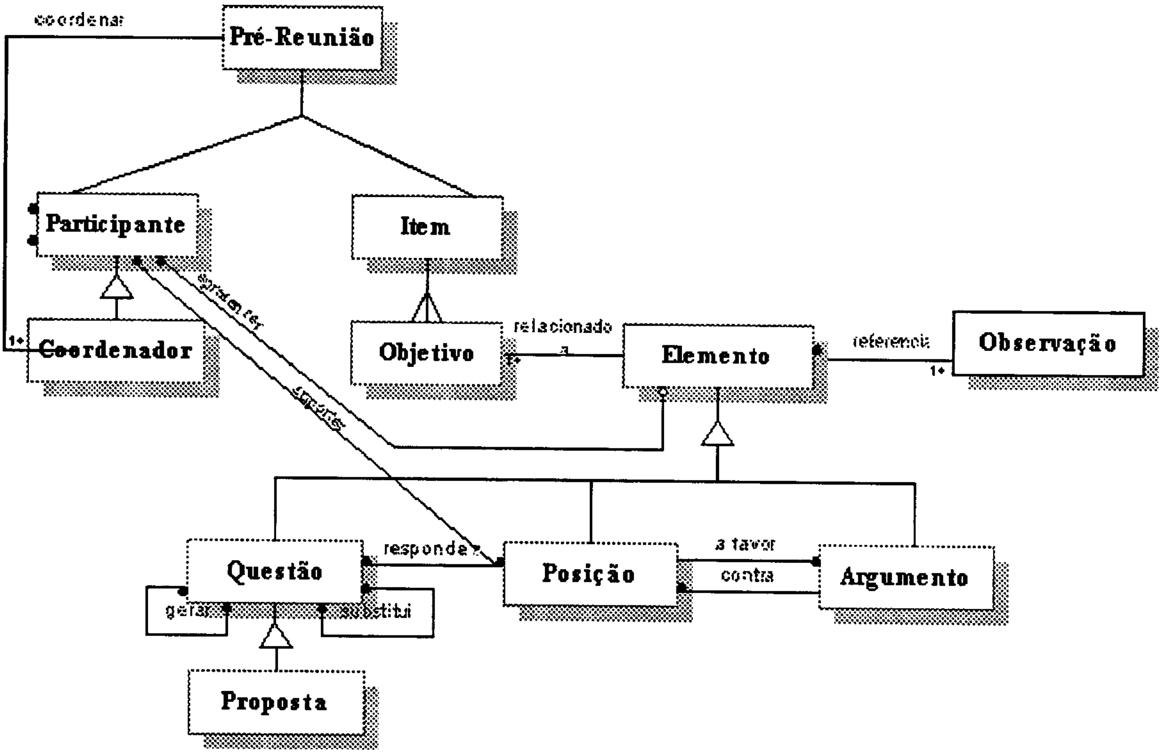


*Figure 3.8: SISCO-Rio Data Model*

### 3.2.2. SISCO-Rio Architecture

SISCO-Rio is a client/server architecture system. The client runs on the most popular Web-browsers available, such as Netscape 2.0 or superior, and Internet Explorer 3.0 or superior. The server side, needs a Web server and a relational DataBase Management System (DBMS) SQL compliant.

SISCO-Rio server is composed by a CGI (Common Gateway Interface) program which provides the bridge between the Web-server and the DBMS server. For each data request, the CGI program connects to the DBMS server, and sends one or more embedded SQL queries. The result of each query is formatted in the HTML format, and sent back to the Web client. In order to diminish the amount of data traffic concerning the Interface, and also to provide some client functionality, like the pre-evaluation of the data entered by the user, the JavaScript language was added up to the HTML code. Therefore, the CGI program, not only builds the HTML output interface of the data extracted from the DBMS, but also embeds in it some JavaScript code, which will be interpreted by the Web browser, at the client side.

The development platform for the SISCO-Rio running prototype, was composed by client machines running a Netscape or Microsoft 3.x web browser, and two different Sun/OS machines on the server side. SISCO-Rio runs on top of the NCSA 1.4 web server, and of the CA OpenIngres 1.1/04 DBMS. The Web server, the SISCO-Rio CGI and the DBMS client runs on one of the Sun machines called "borel", while the DBMS server runs on the other Sun machine called "leme". Figure 3.9 illustrates SISCO-Rio development platform.

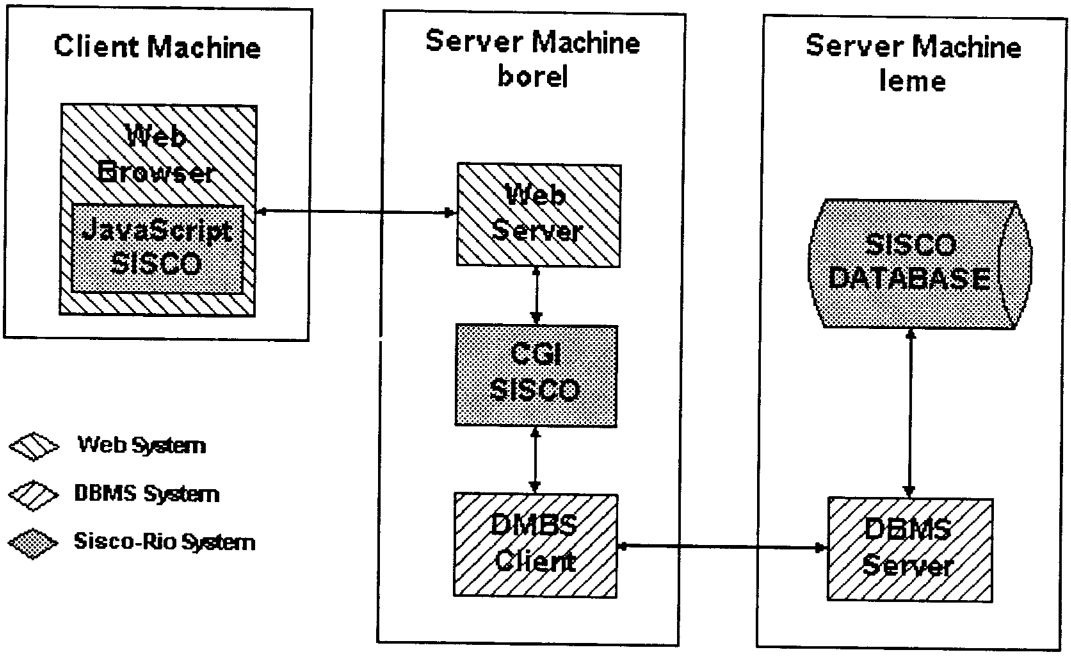
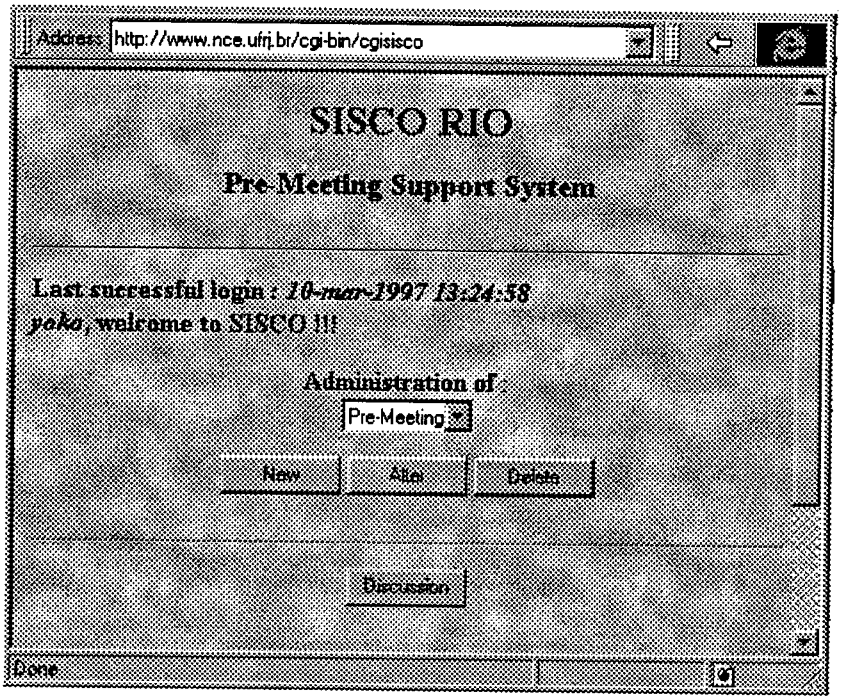


**Figure 3.9: SISCO-Rio Development Platform**

### 3.2.3. SISCO-Rio Prototype

As mentioned in the previous section, SISCO-RIO prototype implements two basic functionality. The first one deals with the administrative part of the system, which is responsible for the creation and maintenance of a pre-meeting and its respective participants, items and objectives. Only coordinators have access to this option, but in this version of the system, anyone can become a coordinator, once he/she creates a new pre-meeting. The second way of using SISCO-RIO is to take part on a pre-meeting, where the users discuss their ideas through the contribution of key elements such as issues, positions, arguments and remarks, following a pre-defined hierarchic structure.

**Administrating a Pre-meeting:** The SISCO-RIO first page asks the user for a login/password. After logging in, the user gets the main page of the system (see figure 3.10). The NEW button takes the user to a new page (see figure 3.11) where he/she may choose among the pre-meetings of which he/she is the coordinator. This page provides all the functionality for creating a new pre-meeting, its participants, items and objectives. For modifying or deleting a pre-meeting, item, objective or participant, the user should go back to the main administration page showed in figure 3.10, and hit on the correspondent button, ALTER or DELETE.



Address: http://www.nce.ufrj.br/cgi-bin/cgisisco

SISCO RIO

Pre-Meeting Support System

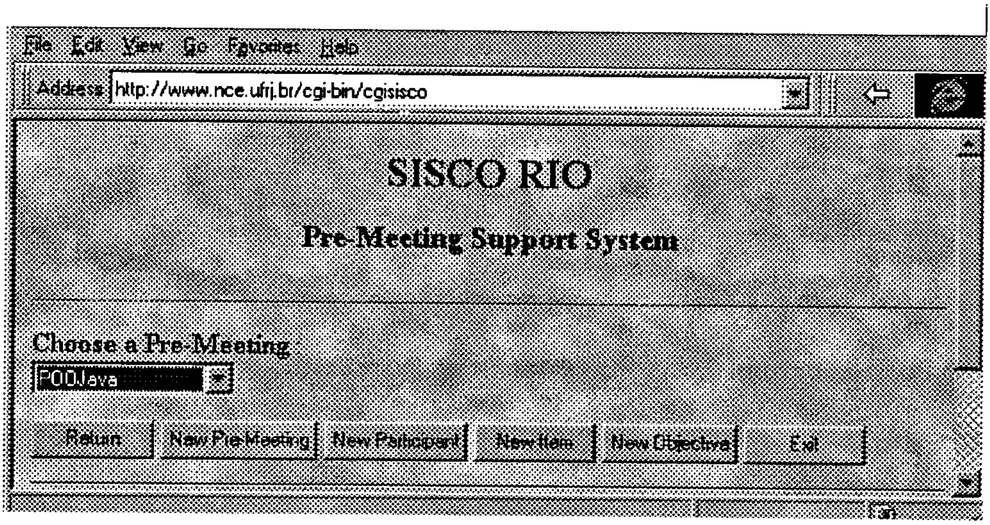
Last successful login : *10-mar-1997 13:24:58*

*yoko*, welcome to SISCO !!!

Administration of :

Pre-Meeting

*Figure 3.10: SISCO-Rio Main Page*



Address: http://www.nce.ufrj.br/cgi-bin/cgisisco

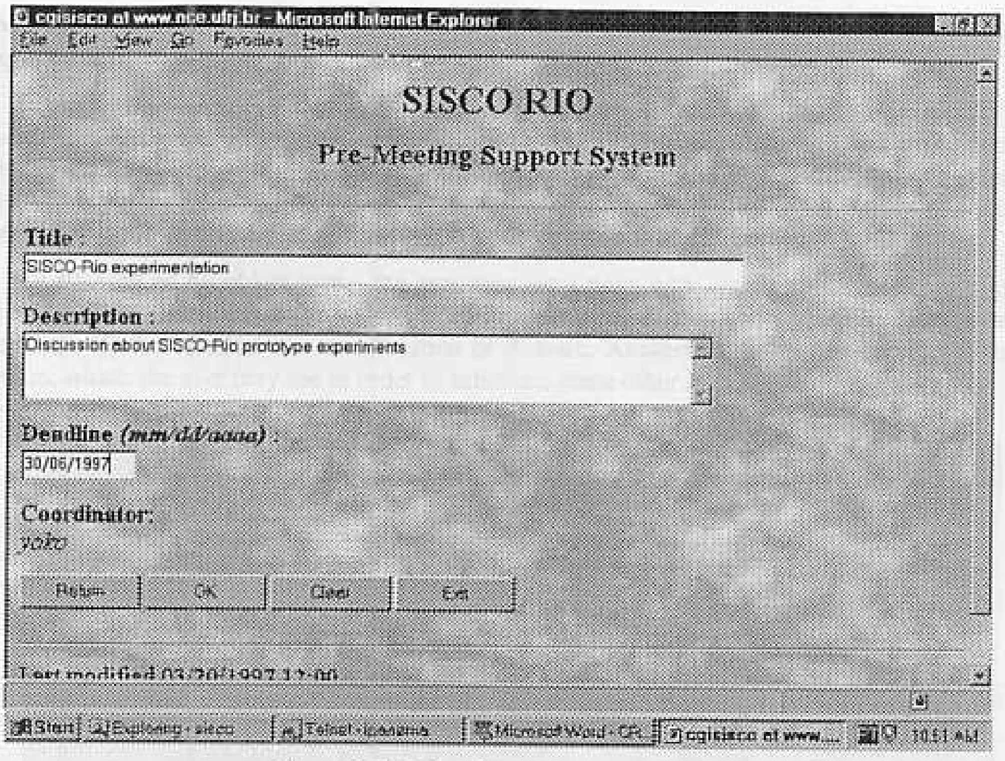
SISCO RIO

Pre-Meeting Support System

Choose a Pre-Meeting

POOJava

*Figure 3.11: Creating New Pre-Meetings, Participants, Items and Objectives*


cgisisco at www.nce.ufrj.br - Microsoft Internet Explorer
File Edit View Go Favorites Help
Start
Exploring - sisco
Telnet - ipanema
Microsoft Word - CR
cgisisco at www....
10:51 AM

SISCO RIO

Pre-Meeting Support System

Title : SISCO-Rio experimentation

Description : Discussion about SISCO-Rio prototype experiments

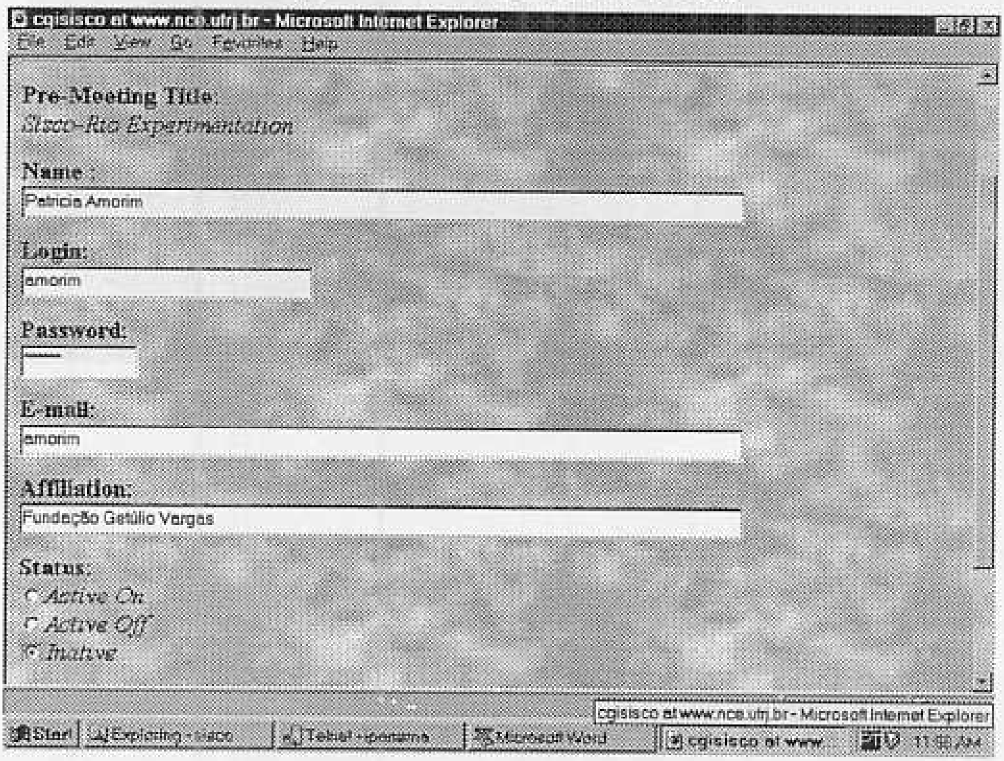
Deadline *(mm/dd/aaaa)* : 30/06/1997

Coordinator: *yoko*

Return OK Clear Exit

Last modified 03/20/1997 12:00

*Figure 3.12: Creating a new pre-meeting*


cgisisco at www.nce.ufrj.br - Microsoft Internet Explorer
File Edit View Go Favorites Help
cgisisco at www.nce.ufrj.br - Microsoft Internet Explorer
Start
Exploring - sisco
Telnet - ipanema
Microsoft Word
cgisisco at www...
11:00 AM

Pre-Meeting Title: *Sisco-Rio Experimentation*

Name : Patricia Amorim

Login: amorim

Password: ******

E-mail: amorim

Affiliation: Fundação Getúlio Vargas

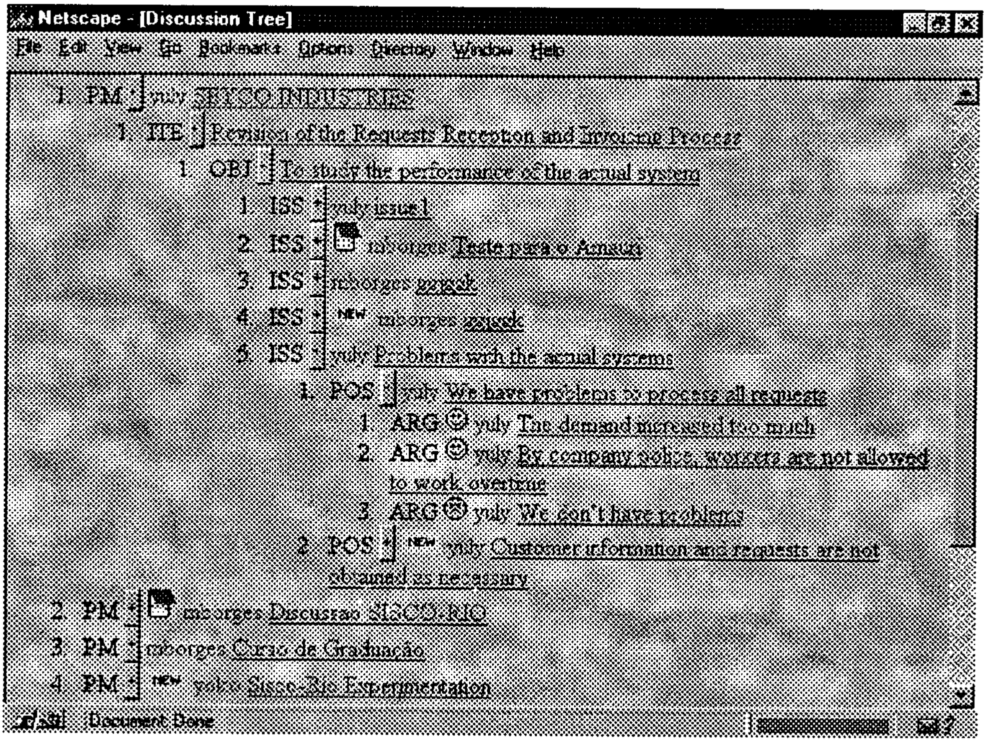
Status:
○ *Active On*
○ *Active Off*
● *Inative*

*Figure 3.13: Creating a new participant*

Figures 3.12 and 3.13 show the system screens where it is possible to create a new pre-meeting and a new participant, respectively. As items are subordinated to pre-meetings, their creation, modification or deletion demands the selection of the pre-meeting they belong to. Analogously, objectives are subordinated to items, which means it is necessary to go through the selection of an item after selecting the pre-meeting.

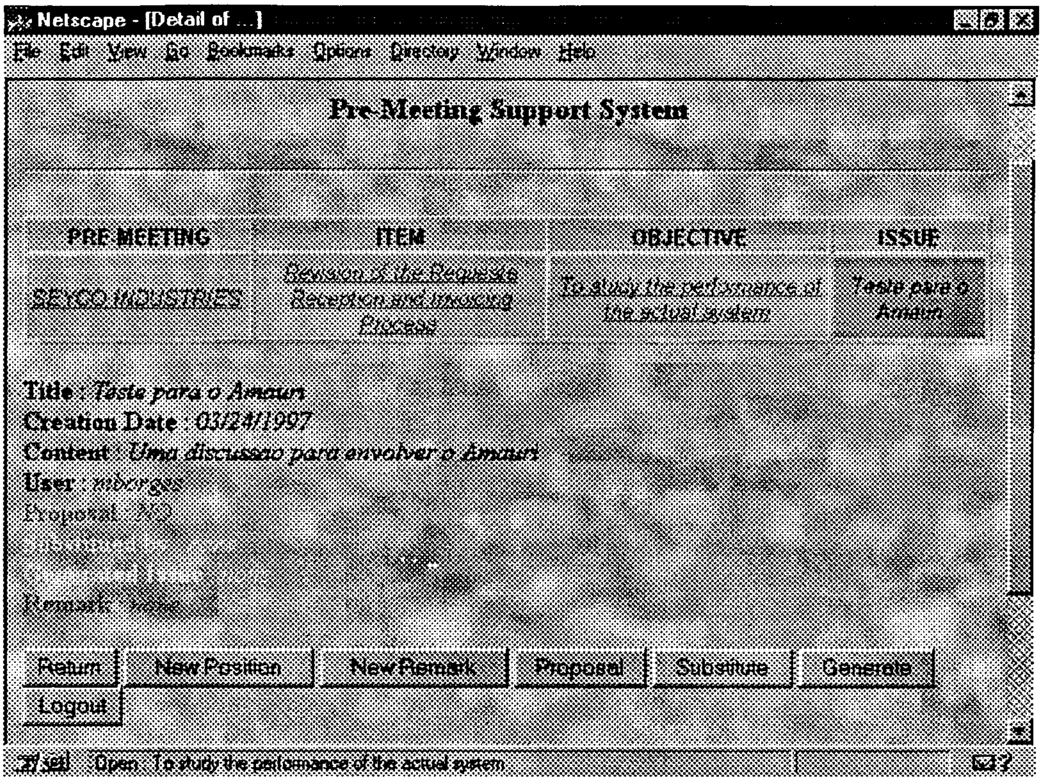
**Participating on a Discussion:** In order to participate on a discussion, once in SISCO's main page (see figure 3.10), the user should choose the DISCUSSION button. Then, the user gets a hierarchic tree (see figure 3.14) containing all the pre-meetings of which he/she is a participant. The hierarchic structure will expand or contract by using the plus/minus buttons (+/-). The top most level lists the pre-meetings, which are followed by items, objectives, issues, positions and arguments.

By clicking on an object name, the user may get a page with details about that object, and also some specific functionality related to it. Figure 3.15 shows a page with details about an Issue, where it is possible to ask for the creation of a new Position or Remark. Another available functionality is the Issue Substitution, which the user may use in order to substitute some other Issues.


Netscape - [Discussion Tree]
File Edit View Go Bookmarks Options Directory Window Help
Document Done

```
1. PM - yuly SISCO INDUSTRIES
   1. ITE - Revision of the Requests Reception and Invoicing Process
      1. OBJ - To study the performance of the actual system
         1. ISS + yuly issue1
         2. ISS + mborges Teste para o Amauri
         3. ISS + mborges [illegible]
         4. ISS + NEW mborges [illegible]
         5. ISS - yuly Problems with the actual systems
            1. POS - yuly We have problems to process all requests
               1 ARG ☺ yuly The demand increased too much
               2. ARG ☺ yuly By company police, workers are not allowed
               to work overtime
               3. ARG ☹ yuly We don't have problems
            2. POS + NEW yuly Customer information and requests are not
            obtained as necessary
2. PM + mborges Discussao SISCO-RIO
3. PM + mborges Curso de Graduação
4. PM + NEW yuly Sisco-Rio Experimentation
```

*Figure 3.14: Discussion hierarchy page*


Netscape - [Detail of ...]
File Edit View Go Bookmarks Options Directory Window Help
Return
New Position
New Remark
Proposal
Substitute
Generate
Logout
Open: To study the performance of the actual system

Pre-Meeting Support System

| PRE-MEETING | ITEM | OBJECTIVE | ISSUE |
|---|---|---|---|
| SEXCO INDUSTRIES | Revision of the Requests Reception and invoicing Process | To study the performance of the actual system | Teste para o Amaun |

**Title** : *Teste para o Amaun*
**Creation Date** : *03/24/1997*
**Content** : *Uma discussao para envolver o Amaun*
**User** : *mborges*
Proposal : NO
[illegible]
[illegible]
Remark : none

*Figure 3.15: Issue details page*

SISCO-Rio prototype is already available for test and experimentation at the web address http://www-aep.nce.ufrj.br/sisco.

## 3.3. SISCO-Recife

SISCO-Recife prototype is now the main subject of a Master thesis dissertation at the Departamento de Informática of the Federal University of Pernambuco. SISCO-Recife development is foccused on enhancing of the original SISCO data model specification. So far, the Recife team has been dedicated to develop the SISCO-Recife model, and to validate the usability of the platform chosen. SISCO-Recife prototype is not yet available and operational. For this reason it was not possible to show pictures of the prototype interface. However we present its architecture and platform which are already defined.

### 3.3.1. SISCO-Recife Architecture

SISCO-Recife is a client/server architecture system. The client runs on the most popular Web browsers available, such as Netscape 2.0 or superior, and Internet Explorer 3.0 or superior. The server side, needs a relational DataBase Management System (DBMS) SQL compliant.

SISCO-Recife server is composed by a CGI (Common Gateway Interface) program which provides the bridge between the Java Applets and the DBMS server. For each data request, the CGI program connects to the DBMS server, and sends one or more embeded SQL queries. The result of each query is formatted in the PLAIN format, and sent back to the Web client. The application front-end is totally developed using Java language.

The development platform for SISCO-Recife prototype, is composed by Web browsers which run Java Applets on the client side, and a Sun/OS machine on the server side, which runs the Web and de DBMS servers. For developing the java Applets it has been used a Sun/Solaris machine, while for the CGI development it has been used a Sun/OS machine called *ulysses*. Figure 3.16 illustrates SISCO-Recife development platform.

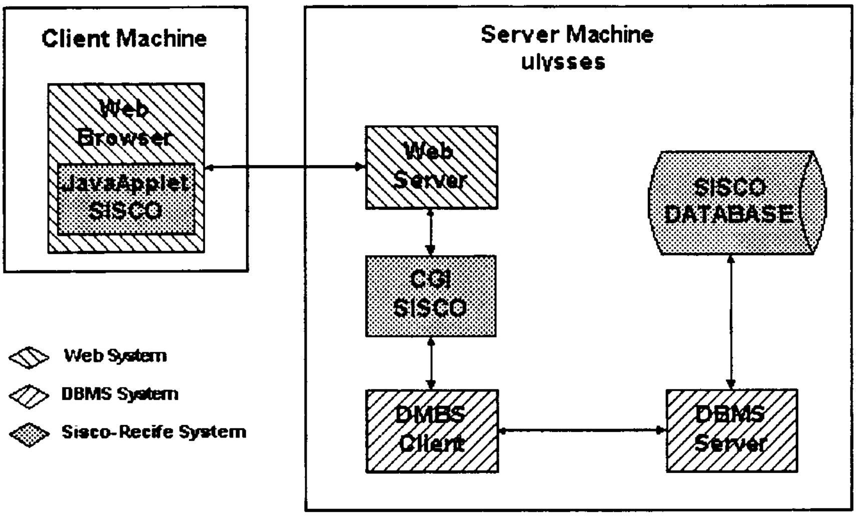


**Figure 3.16: SISCO-Recife Development Platform**

# 4. Implementation Discussion

First of all, it is not our intention to choose one of the prototypes as the best among the others. Instead, our main objective here is to identify what are the best aspects of each prototype platform, and how our development experience was enriched by the exploration of different platforms. The following sections organize the implementation discussion in the relevant topics.

## 4.1. Interface

One of the most evident differences among the prototypes relates to the Interface. Choosing the Web as the development platform may provide a lot of benefits, such as distribution and accessibility, however it does impose some interface limitations. SISCO-Rio and SISCO-Recife found ways out of these limitations by using JavaScript or Java Applets, respectively. Programming Java Applets allows the implementation of much more powerful Interfaces than using JavaScript, however it demands specialized knowledge, such as object oriented programming. On the other hand, even though with weaker Interfaces, it is possible to provide sufficient functionality using JavaScript, with relatively no cost. In fact, JavaScript is very simple, and consequently allows short term implementation.

Java stand alone applications, like USISCO, may provide even more sophisticated interfaces, however USISCO team experienced some limitations dealing with the AWT library. It was necessary to re-implement some interface components in order to provide a basic functionality, such as the change of the mouse icon, when it passes over a "clickable" element.

## 4.2. Group Memory Storage

All the prototypes use a relational DBMS as the SISCO group memory storage, but SISCO-Rio and SISCO-Recife teams conceived their prototypes with emphasis on the use of DBMS's. Both prototypes use SQL-embedded as a way of preserving portability, allowing an easy migration to most of existing relational DBMS's. However, even though the migration is easy, some of them implement only a subset of the SQL standard, which might diminish systems portability. In this sense, USISCO choice might be considered more portable. Different from the other two prototypes, which use very robust and commercial

DBMS's, USISCO uses a simpler and public domain DBMS, which can be replicated and installed in many different platforms, providing the desired portability.

The fact that OODBMS's are beginning to be commercially accepted, and that there are not many choices available, contributed to the decision on using a relational DBMS. However, this decision turned out to be a little costly, because the original SISCO project relies on an object oriented methodology. For this reason, a significant effort was spent on mapping from the OO paradigm to the Relational Model.

### *4.3. Platform*

As we already said, accessibility is one of the advantages of choosing the Web as a client-server development platform. Users may be located at any place in the planet, and since there is an Internet connection, it is possible to use the system. Besides, all an user must know is a URL address, which gets him/her straight to the system main screen. There is no need of time consuming installation programs and problems, the user depends only on the existence of a Web browser on the client machine.

Accessibility brings another issue to focus: maintainability. The web platform provides a very transparent way of distributing new versions of the system. The user hardly notice that he/she is using a new version of the system, and, it is not necessary to request one.

### *4.4. Future versions*

All SISCO prototypes are still not complete. Some of the specified functionality were poorly or not implemented at all, such as awareness notifications, system outputs, pre-decisions, infobase, etc. USISCO implements some notification functionality, identifying new elements, and changing colors when the user reads an element. However, we believe there is much more to be done with this respect, such as the need of an integration with the e-mail system, in order to provide a more efficient user notification.

System output is another very important functionality to be implemented by all the three prototypes. There is no sense in a pre-meeting system which does not provide the outputs required by the follow up activity: the real meeting. Finally, pre-decisions and Infobase functionality need to be reviewed for a complete and efficient implementation, however they are not mandatory for these first prototypes.

After implementing what agreed to be necessary for a first SISCO validation, all the prototypes must go over a test phase. Knowing the problems of client-server running platform, it is suggested to put the prototypes under extreme tests, such as energy faults, maximum expected number of concurrent users, etc.

## 5. Conclusion

The next step to the three SISCO prototypes described here is their use and consequent SISCO validation. This will include not only functional and conceptual issues, but also performance and security issues. Performance measurements probably will cover some important aspects, like network traffic overload between the client and the server machines, server and client processing overload.

Security is certainly another issue that should be discussed. Because of the high costs of implementing security functionality, all the prototypes did not include this part. On the other hand, the prototypes do not implement it, we believe it is more important to validate SISCO basic functionality first, and then evaluate security matters on a next version.

The SISCO "testbed" groups will be chosen and observed carefully. It is now one of the major concerns of SISCO managers, finding the perfect testbed groups. Users profile, motivation, and organization are some of the aspects to be considered in the search for a testbed group. Also, monitoring utilities will be needed in order to allow users behavior capture while they are using SISCO prototypes.

There is no doubt that we had profit with the prototype platform diversity. We have been able to share different implementation experiences, exchanging suggestions and solutions to our individual implementation problems, and envisioning the suitability of each platform. We believe this experience will contribute largely for the construction of a definite, adequate and enhanced version of a SISCO system.

## 6. Acknowledgements

The SISCO project is sponsored by CYTED, Subprogram VII. SISCO-Rio development was partially sponsored by CNPq under grant # 28374692. USISCO development was partially supported by grant No.1970342 from FONDECYT (Chile). The university departments, respectively NCE at UFRJ, DCC at U. Chile and DI at UFPE provided the necessary local support for the project development.

We would like to thank our advisors Marcos Borges, Jose Pino and Ana Carolina Salgado for the opportunity of taking part on such an interesting Project. In particular, SISCO-Rio authors would like to thank Maria Luiza Campos, Patrícia Amorim and Suzana Cesar for their participation on SISCO-Rio prototype development. USISCO team especially thanks professor Nancy Hitschfeld for her greatly useful advises and for her patience.

## References

[Bellassai et al. 95] "SISCO: A tool to improve Meeting Productivity" - Bellassai, G. & Borges, M. & Fuller, D. & Pino, J. & Salgado, A. - Proceedings of CYTED-RITOS International Workshop on Groupware - CRIWG'95, INESC, Lisbon, Portugal, 1995;

[Bellassai et al. 96] "An IBIS-based model to support group discussions" - Bellassai, G., Borges, M., Fuller, D. A., Pino, J.A. and Salgado, A.C.. In Proceedings of the 1996 IFIP International Workshop on the Office of the Future, Tucson, Arizona, April 1996.

[Borges et al. 95] "Trabalho Cooperativo Suportado por Computador" - Borges, M. & Campos, M. & Cavalcanti, M. - XV Congresso da Sociedade Brasileira de Computação - Jornada de Atualização em Informática - Canela, RS - Brasil, 1995;

[Conklin et al. 88] "gIBIS: A hypertext tool for exploratory policy discussion" - Conklin, J. & Begeman, M. - ACM Transactions on Office Information Systems, 6 (3), October 1988, pp 303-331.

[Daft & Lengel 86] "Organizational Information Requirements, Media Richness and Structural Design" - Daft, R.L. and Lengel, R.H., Management Science, No. 5, May 1986, pp. 554-571.

[Ellis et al. 91] "Groupware some issues and Experiences" - Ellis, C. & Gibbs, S. & Rein, G. - Communications of the ACM, vol 34, No 1, Januar, 1991;

[Kunz et al. 70] "Issues as Elements of Information System" - Kunz, W. & Rittel, H. - Working Paper # 131, Institute of Urban and Regional Development, University of California at Berkeley, 1970;

[Nunamaker et al. 91] "Electronic Meeting Systems to Support Group Work" - Nunamaker, J. & Dennis, A. & Valacich, J. & Vogel, D. & George, J. - Communications of the ACM, Vol 34, No 7, July, 1991;

[Watson 93] "Yin and Yang, Social Forces, and Meeting Design" - Watson, R. T. - In: "Local area network applications: leveragin the LAN" - Vogel, D. & Marshall, P. & Glasson, B. - Verrijn-Stewart, A. A., eds. Amsterdam: North-Holland, 1993;